Preliminary Literature Review of Policy Engineering Methods - Toward Responsibility Concept

Feltus, Christophe

Published in:
Proceeding of 3rd international conference on information and communication technologies : from theory to applications (ICTTA 08), Damascus, Syria

DOI:
10.1109/ICTTA.2008.4529912

Publication date:
2008

Document Version
Early version, also known as pre-print

Preliminary Literature Review of Policy Engineering Methods

Toward Responsibility Concept

Christophe Feltus
Centre for IT Innovation
Public Research Centre Henri Tudor
29, Avenue John F. Kennedy, L-1855 Luxemburg
christophe.feltus@tudor.lu

Abstract— **INTRODUCTION**

This paper introduces a preliminary review of the research currently performed in the field of Policy. This review aims to understand the approaches covered by main research streams in that area and to highlight the advantages of the essential and most renowned solutions. The review of the literature quickly provides a plethora of publications that presents innovative proposals on the matter of policy conceptual model, engineering methods, elicitation languages, as well as cases studies. It also brings out that the papers most often refer rather evasively to the organizational model layers when aligning and positioning their theory with organizational concepts. Consequently, it sounds useful to orient and improve our own developments in the purpose of ameliorate that issue.

Based on that overview's results, we are able to orient our researches more deeply by proposing an innovative approach that focuses in one hand on a policy model designed to take into account the responsibility of stakeholders and in the other hand on policy engineering method that takes care of business process while at the same time using requirement engineering principles. Responsibility is a notion that remains rarely addressed and that however embodies important and well-know concepts like accountability, capability and commitment. Moreover, responsibility constitutes a fundamental notion of management theory and is consequently identified as a meaningful bridge toward organizational artifacts. Exploiting process to define policy seems likewise to offer new research opportunities since process organizations become a more widely spread structured approach.

Keywords- Policy concept; Responsibility; Corporate Governance; Right management; Business requirement; IT Governance

I. INTRODUCTION

It is notable that nowadays, an aspect for a long time remained overshadowed appears to be from a major interest. This aspect is the responsibility committed from a person to perform a task. This responsibility is often perceived as a combination of rights and obligations. However, current business in financial sector for instance demonstrates that the moral aspect is improvable and that taking care of that matter would avoid in some cases malfunctions of the system. Our work starts based on the hypotheses that this responsibility is composed by the tuple {Capability, Accountability, and Commitment}. Our previous work [1] has introduced principal semantic characteristics about those three concepts and has brought formalizing elements using standard logical.

It is rapidly observable when beginning to launch into policy literature that a very large amount of authors shows interest in that concern. Whatever that meaningful proliferation of works and states of the art with regard to it, it is noteworthy that up to now it doesn't really exist some distinction between works addressing access control model, policy model, role engineering and permission/policy engineering. Based on that assumption, it appears substantial for apprehending that topic to clarify this point and to highlight the existing dichotomy between model and method. To perform our review, we will base our analysis on a commonly accepted idea that a model or conceptual model is a representation designed to show the structure of a system or concept and that (at least in our case), a method is a body of techniques for collecting data necessary to instantiate the conceptual model. Consequently and as illustration, the Role-Based Access Control (RBAC) model [2] proposes a structure for providing access based on role whereas role engineering [3] and [4] is a method aiming to define roles to instantiate the conceptual model. Identically, policy may also be modeled and it exists a proliferation of methods to instantiate it. These methods may be classified according to the technique they use. We propose to start with methods based on RE and to continue with a list of others. Moreover, it is more frequent to read paper targeting policy language than policy model. Those policy languages are innumerable and spread over the entire organizational model layers. Most famous of them are Ponder [5], Policy Description Language [6], Security Policy Language [7], and Rei [8]. Amazingly, the policy model used to support the policy expression by the policy language remains rarely specified.

The next section introduces Camerer's observations over researches in the domain of policy, section III reviews the concepts of responsibility in access control models and section

IV reviews the same concepts in engineering methods. Section V concludes and presents future works.

II. FROM BUSINESS TO SECURITY POLICY

Before going ahead in the literature review, let make a hook to understand the analysis made by Camerer [9] on researches in business policy and strategy. An important observation in its work is that: « *There are at least three symptoms of the disease causing the queasy dissatisfaction with policy research:*

a) Concepts are often ambiguous and their definitions are not agreed upon;

b) Checklists or theories are rarely tested, and never tested directly against competing theories and

c) Theories do not 'cumulate' or built upon previous theories as they should.

These three deficiencies are a result of the way policy research is typically done."

Camerer explains that policy research should evolve from an inductive to a deductive approach. He argues that induction contribute to an unproductive debate about variable definitions and to a lack of testability and failure of theory. Unlikely, his conviction is that deductive models can express hypotheses in a language that is more amenable to progressive debate. This point of view is a precious warning we have to take into account before beginning our researcher in that it may prevent us to perpetrate the same mistakes. This warning is moreover substantial because of the still subjective character of the moral aspect under focus. In his work, Camerer only addresses business policy. Therefore, this consideration needs to be adapted according to our research's context and it is consequently necessary to clarify the relation that exists between business policies and IT policies. Wies [10] shows the links between high and low-level policies. He depicts the variation of importance of the technology and the business aspects when translating high-level onto low-level policies. High-level policies tend to focus on business aspects whereas low-level policies focus on technology aspects. Although they are spread on different abstraction layers of the policy hierarchy, business policies and IT policies are consistent because both are derived from (management and/or IT) goals and hence embody (management and/or IT) strategy's aspects. Rifaut et al. [11] propose to use GORE methods to define goals, strategies and policies. Rifaut explains that these methods can be used to analyse and model systems at all organizational level, from business models up to architectures, see Figure 1.

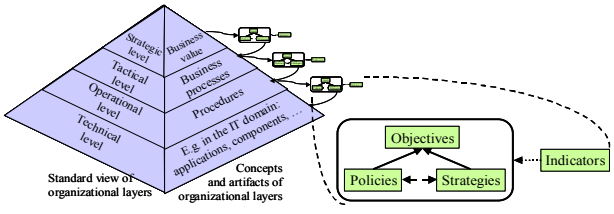

Figure 1. GORE model for policy refinement

Based on the previous assumption that it exists links between policies from different layers, further analysis of the literature has been conducted to depict the principal elements that compose the policy concept.

III. RESPONSIBILITY AND ACCESS CONTROL MODELS

The state of the art of policy's concepts introduces a review of 4 main recognized access control models: Mandatory Access Control (MAC), Discretionary Access Control (DAC), Role-based Access Control (RBAC) and Usage Control Model (UCON).

MAC is designed with the characteristic that one authority fixes the access right rules and that users are not permitted to modify them. MAC defines objects and subjects. Each objects and subjects are classified into classes and levels. Objects are resources to be protected whereas subjects are active entities that access objects. Lattice-based access control is a type of MAC policy. In that control, each access class is associated a security level and a set of category. The security level determines the level of sensibility of objects and subjects. I.e.: TopSecret>Secret>Confidential>Unclassified. The set of category corresponds to an area of competence or to a function. I.e.: Army, Navy, Nuclear, Administration. Access is allowed if subject clearance level >= object sensitivity level. The Bell-LaPadula model has been developed in the same time that MAC and focuses on data confidentiality. The system is composed of objects, subjects and actions. Each object is associated to an access class that defined its level of sensibility and each subject is associated to a clearance. Subjects may exercise actions on objects: I.e. read-write [12]. This access control is mainly based on the concept of right. Subjects are not allowed to modify rules and both obligation and commitment of subject are not addressed.

In DAC users may receive possibilities to define their own AC rules on some specific objects. In general, users are identified and it exists rules that determine who is allowed to perform what on which resources. One main recognized form of DAC is the access matrix. This matrix as been formalized by Harrison, Ruzzo, and Ullman through the HRU model that defines 6 types of primitive operations: enter or delete an action in the matrix, create or delete an object or a subject. The access matrix may however become difficult to managed and encompasses a big number of free cells that makes the matrix ineffective. Three others approaches have therefore been proposed: authorization tables, Access Control Lists (ACL), and capabilities. Authorization tables are composed of tuples {user, privilege and object} and are generally used for database management system. Those tables permit to reduce the size of the matrix. With ACL, an object is associated to a list of privileges that each subject have on object (I.e. unix: rwx r-x rw-). Finally, capabilities are represented by a list associated to each subject and that contains its access right towards objects. DAC also not addresses commitment and obligation.

RBAC policy is a model that permits to effectively align access rights with the organizational structure of the company. RBAC is based on the principle that the most important information to access a resource is the role played by a user within the system [2]. A role is defined in RBAC as "a job

function within the context of an organization with some associated semantics regarding the authority and the responsibility conferred on the user assigned to the role". That means that a user is associated to a role and that permissions are also associated to that role.

RBAC model is a junction of many models: core RBAC (RBAC0), hierarchical RBAC (RBAC1), constrains RBAC (RBAC2) (static separation of duty relations and dynamic separation of duty relations) and constrains RBAC with role hierarchies (RBAC3). Core RBAC encompasses following elements: users (USERS) (human or sometime processes), roles (ROLES), objects (OBS) that are access resources, operations (OPS) that are processes that execute functions in the name of users and permissions (PRMS) that are authorizations to access system's objects and that consequently establish links between objects and operations.

The core RBAC also defines two kinds of relations between these elements: Firstly, user assignment (UA) that represent the relation between users and roles. Moreover a user may be affected to one or more roles and a role may encompass many users. Secondly, permission assignment (PA) defines the relation between roles and permissions. In the same way, much permission may be affected to many roles. Moreover, users-roles connections are established by the sessions (SESSION). RBAC addresses the capability (AC), obligation is introduced by RBAC2 that imposes some rules for accessing objects and commitment remains not addressed.

Park et al. [2] have introduced UCON in 2002. The term "usage" means the usage of rights upon digital objects. UCON joints in a unique model traditional access control as MAC, DAC and RBAC, trust management, and digital rights management (DRM). As explained previously, the traditional AC represents the control in a closed system where users are identified. The trust management system is used to assign authorization to unidentified subjects in an open environment like Internet. DRM assures access control to digital data and in that, the control is assured at the client-side. These three models are complementary and target different objectives. Park et al. argue that needs evolve and consequently that in some situations the usage of the three models together is justified and is made possible thanks to UCON. The model UCON encompasses Authorization, Obligation and Condition. Authorizations are functional attributes that must be evaluated before a usage decision and that return to the subject whether yes or no the access is granted over the object. Obligations are functional attributes that allow verifying if the subject has satisfied some conditions before and during the usage of the object. Conditions are decisional factors based on the environment or the system. The advantages of UCON are firstly that the model proposes a possibility of ongoing decision. That means that decision is taken before and during the usage. Secondly, the mutability of attributes that is an update of subject or object attributes after or during the usage.

Our overview has also covered others approaches that due to the size of the paper are not presented here. In summary we may observe that firstly, some concepts are commonly accepted, such as right, role and obligation. Definition of the two firsts concepts are scarce. Only one definition has been found for the concept of "right": the right (or permission) is explicitly granted to a subject to access an object in a specific mode, such as read or write [2]. For the concept of "role", only one definition has also been found in **Error! Reference source not found.**. The concept of obligation is subject to more debate. For Bettini et al. [14], obligations are conditions or actions that must be fulfilled either by the users or the system after the decision. In [2], Sandhu et al. define obligations as requirements that have to be fulfilled by the subject for allowing access. Crook et al. [15] extend the notion of obligation to obligation policy that relate to actions that must be carried out on targets by subjects when a predefined event occurs and Haley et al. in **Error! Reference source not found.** define it as what actions must be taken before access can be granted.

TABLE I. AC MODEL AND RESPONSIBILITY COMPONENTS

	MAC	DAC	RBAC	UCON
Subject	Yes	Yes	Yes	Yes
Object	Yes	Yes	Yes	Yes
Group	No	User Group	Role	Defined by objects and subject's attributes
Capability	Access Right	Access Right	Access Right	Access Right
Accountability (Obligation, Constraint)	No	No	Yes, static et dynamic separation of duty	Defined by objects and subject's attributes
Commitment	No	No	No	No

IV. RESPONSIBILITY AND ENGINEERING METHODS

This second part aims at analyzing some recognized engineering methods with the same objective to identify how they address responsibility. Even if number of work has already been produced to overview researches achieved in the domain of security policies requirement engineering [15][17][18] and **Error! Reference source not found.** none has targeted the responsibility through the tuple {Capability, Accountability and Commitment}. This section of the overview is focused on policy engineering methods that use software requirement engineering methods for defining policy requirements. The requirement engineering considers the functional requirement and the « quality » requirement (or no-functional).

KAOS is a goal-oriented software requirement engineering approach that allows calculating requirements from goal diagrams. KAOS stands for Knowledge Acquisition in automated specification or Keep All Objects Satisfied. KAOS permits to specify high-level requirements and defines a set of "meta-concepts" and "meta-relationships". Some of the meta-concepts may be used to AC. The agent component is an important one and is defined as either human beings or automated components that are responsible for achieving requirements and expectations. To achieve the requirement, the agent has capabilities. KAOS defines the responsibility as the relationship that connects an agent to a requirement for which the agent is responsible. Fontaine [20] has used KAOS to

refine security needs in authorization rules and in security policies. Whatever, KAOS is not the solution to design all kinds of policy such as for example delegation policy.

The i* framework is an agent-oriented modeling framework, Yu et al. [21], that supports goal-oriented strategic modeling and analysis of requirements by using three mains concepts that are: actors, intentional elements, and links. Actors are described in their organizational setting and have attributes such as goals, abilities, beliefs, and commitments. Actors can be agents, roles, and positions. Agents are concrete actors, systems or humans, with specific capabilities. It was initially used to analyze and model business processes and now, it evolves to the development of model for security and privacy requirements. I* doesn't provide any method to identify and define roles and permissions but it allows to model relations between actors. More particularly, Liu et al. demonstrate in [22] how it is possible to derive AC restriction when using the actor boundary in a Strategic Rationale (SR) model. The SD diagram represents the strategic dependencies of the actors. A dependency is an "agreement" between 2 actors.

Goal-Based Requirements Analysis Method (GBRAM) [23] aims at representing a system and its environment as a collection of agents. The agent has to achieve a goal based on the assumed responsibility for it.

Crook et al. propose in 2003 [15][17] and [24] a framework named Analytical Role Modeling Framework (ARMF) for modeling roles according to the RBAC model. The particularity of its approach is that it is based on the Mintzberg theory [25] that classifies roles according to three categories that he integrates in the framework. Those categories issue from the organizational model are: « Roles based on seniority », « Roles based on function » and « Roles based on market ». ARMF encompasses 2 levels: a « meta-level » that include roles-types, asset category, and context-types. Those key conceptual components permit to define access policies. The second level is the instance level that provides instantiated answers about users, context, assets and roles (types: «functional», «seniority» and «contextual»). Unlike requirement engineering methods such as KAOS, GBRAM and methods based on Uses Cases, that only take into account definitions of actors or agents, and in addition to the i* framework that only addresses one differentiation between different type of roles (roles based on "position" and based on "task" to accomplish), ARMF introduces a context, contextual roles, a hierarchy between assets and a hierarchy between roles. Moreover, ARMF permits to present contextual role by establishing links between the asset and the context and between the context and the role. In the example included in [17], Crook also uses Formal Tropos to prove that its framework to model security policies respects the principal of minimum privilege.

Qingfeng et al. [26] 's framework is named (Requirements-level AC Analysis Framework (RACAF) and defined 4 types of analysis that each addresses different requirement of the access control. The first analysis is a « Data Analysis » that permits to obtain information that must be collected by the system, privacy preferences over that data, the type of data and the data that must be protected by access control. The second analysis is a « Goal/Scenario-Based Task Analysis ». This analysis permits to identify the task to perform and derive from it the objectives, permissions, obligations and the context. From the scenario analysis, it is also possible to extract event sequences that include actors, actions, permissions, pre-conditions and post-condition. The third analysis is the « Organizational Structure Analysis ». This analysis' target is to identify the existing relationship between actors and consequently the organizational hierarchy, the delegation and the roles. The last analysis is the «Information Flow Analysis» that is interesting because it analyzes the flow of information outside the company. The method proposed by Qingfeng begins with a Non-Functional Requirement (NFR) analysis. NFR are assimilated to soft goals, it means that those goals must be satisfied to a certain limit rather than absolutely having to be satisfied. The NFR framework defines the design of the system, whatever access control is treated as a solution to answer soft goals of confidentiality. Further development of Qingfeng's work [26] has led to a goal-driven framework for modeling privacy requirements in the role engineering process. The aim of this 2 phases approach is to make a bridge between high-level privacy requirements and low-level access control policy. The first phase is the Role Permission Engineering (RPE) during which the business processes and tasks are analyzed by applying goal and scenario oriented requirement engineering. The results of this phase are roles and permissions both are compatible with the RBAC model. The second phase is the Role Permission Refinement (RPR) is a refining phase of role and permission according to the organizational structure, policy statement, etc.

Gustaf Neumann et al. have based their work on a method of role engineering based on the scenario-driven techniques [3]. A scenario is a part of a task that is part of a "work profile". Moreover the scenario may be considered as a set of step on which are associated operations with specifics accesses. A subject that performs a scenario must consequently possess all permissions necessary to perform each step of the scenario. Seven main activities are identified by Neumann et al.: Identify and model usage scenarios, Derive permissions from scenarios, Identify constraints, Refine scenario model, Define tasks and work profiles, Derive preliminary role-hierarchy, Define RBAC Model. A set of documents is issued from these activities. I.e.: scenario model, permission catalog, constraint catalog, task definitions, work profile and the RBAC model. This RBAC model is the final result of the role engineering and encompasses all the roles of the system organized in one or more hierarchy. The role finding method proposed declines the responsibility through permissions that are derived from scenario and through constraints to be enforced on permissions. These constraints are however technical and need to be managed by the system, i.e.: separation of duties or cardinalities. They may not be considered as responsibility's constructs.

Another approach lies in using uses cases. In its paper [27], Fernandez et al. explain that an existing method to determine the functional requirement is the usage of uses cases. Users of the system are interviewed in order to express the way they interact with the system. Fernandez proposes a method to determine need for a role considering the uses case and a

security administrator defines authorization rules based on all the uses cases over the system. Uses cases are described with a title, actors (that could be roles, users, or other systems), pre-conditions, descriptions, exceptions, and post-conditions. Access rights to specific object are thereafter declined from uses cases. Theses access rights are then translated under the following format: (S, O, T, P) for respectively (subject, Object, Type of authorized access, optional constraint). Uses cases only permit to define functional specifications. Consequently Fernandez proposes uses cases extension with the aim of taken into account non-functional requirement (such as the security). To achieve that, he uses stereotype, in other words UML meta-classification elements. Within these elements, there are the fault tolerance or the security. Actors deduct rights associated to roles from uses cases considering methods that have to be invoked. Therefore, in a scenario diagram, when an actor interacts with an object by a method, than the use case provide a set of access rights. The access right such as the one defined by Fernandez is represented by the tuple R (A, M, O) with A: the actor, O: the object, and M: the method. For Fernandez, some tools such as Paradigm+ or Rational Rose that are developed to present use cases could be extended in a way to generate needed authorization rules.

Pete A. Epstein [18] proposes a model named Role/Permission Assignment Model (RPAM) that allows decomposing and aggregating role attribution to permission. Epstein based his work on three approaches. The first one is the role-finding approach.

Roeckle et al. [28] propose a process-oriented based approach for defining role. To achieve that, he uses a three-layers meta-model: process, roles and access right. In [29], Roeckle presents its work in progress for role-finding based on a process-based approach.

Chandramouli [29] proposes a framework named Dynamic Authorization Framework for Multiple Authorization Types (DAFMAT) that defines a 5 steps methodology for defining access control service for an information system in the healthcare domain.

The model proposed in [30] is issued from the work of Thomsen et al. The objective of this model is to aggregate different permission to a role by using 3 layers that are: Local Policy, Semantic Policy and Application Policy.

Ponder [5] and [31] is an object oriented policy language for the management of distributed systems and networks. Ponder provide the ability to group policies and structure them to reflect organizational structure and preserve the natural way system administrators operate.

XACML stand for eXtensible Access Control Markup Language and is a policy language that has been specified for access control [32]. It defines a core schema and corresponding namespace for the expression of authorization policies in XML against objects that are themselves identified in XML.

TABLE II. ENGINEERING METHODS AND RESPONSIBILITY'S CONCEPT

	KAOS	I*	GBRAM	ARMF	RACAF	Scenario Driven	Uses Cases
Subject	Agent	Actors	Agent	Users	Actors	Subject	Actors
Object	Yes	Yes	-	Asset	Data	-	Object
Group	-	Yes	-	Yes	Yes	Yes	Yes
Capability (Right, Authorization)	Authorization rules	Abilities and beliefs	-	Permission	Permission	Permission	Access right
Accountability (Obligation, Constraint)	Achieve requirements and expectations	Goal	Achieve a goal	Perform a task	Perform a task	Perform a scenario	Pre-conditions, post-conditions
Commitment	No	Yes	No	No	No	No	No

V. CONCLUSIONS AND FUTURE WORKS

We have analyzed the literature to understand the semantics of AC policy conceptual models and engineering methods. We have observed that some elements are commonly accepted components whereas others remain overshadowed. Subject is the basic component and is most of the time associated to a group. Subject appears as a person, a system or a software component. The most famous group type is the role. Object is also a basic component and could take a large scale of representation. I.e.: the performance of a scenario. Capability is a component that is part of all models and methods. Capability is most frequently declined under access right, authorizations or permissions. Accountability is a component that exists mainly in engineering methods and that is declined as the obligation to achieve a task or to perform an action. Commitment is the most infrequent concept. I* introduces some elements of it (I.e. when defining dependency as an "agreement" between 2 actors) but it remains interpretable if it is a moral concept or an obligation.

Based upon that observation, we state firstly that because the most addressed concern of the capability is the access right, existing models and methods most of the time remain targeting low-level abstract layers of the organization. Secondly, if we consider responsibility as a tuple {Capability, Accountability and Commitment}, we may assert that it doesn't exist nowadays model and method that entirely take into account all responsibility's components.

Consequently, our future works will focus on continuing the development of the model of responsibility, and most specially the concept of commitment that could be the most valuable one when climbing up to the high-level layer of the organizational model.

Another part of our works will aim at defining a new approach for derivate the responsibility from the high-level

down to the lower one. Our first researches demonstrate that potentials solutions are to links responsibility's concepts with organization's processes.

As a conclusion regarding the Camerer's warning of section II, we have done this analysis to clarify the semantic of all components that encompass the responsibility and we may consequently state that symptom a) and c) identified by Camerer has been addressed. Firstly the symptom a) that is "Concepts are often ambiguous and their definitions are not agreed upon" has been partially bypassed with clear literature based enlightenment of the concepts. Secondly symptom c) that is "Theories do not 'cumulate' or built upon previous theories as they should." has been addresses with a tentative definition of "responsibility" considering the way its conceptual component are addresses by others authors.

ACKNOWLEDGMENT

SIM "Secure Identity Management" is an R&D project of the CRP Henri Tudor achieved in collaboration with the University of Luxembourg funded by the National Research Fund Luxembourg. Additionally, I would like to thank Pr. Michaël Petit from the University of Namur for its valuable feedback during the preparation of this paper.

REFERENCES

[1] Christophe Feltus, André Rifaut, An Ontology for Requirements Analysis of Managers' Policies in Financial Institutions, I-ESA2007, Madeira, Portugal.

[2] R. Sandhu, J. Park, Usage Control: A Vision for Next Generation Access Control, The Second International Workshop on Mathematical Methods, Models and Architectures for Computer Networks Security, 2003.

[3] Gustaf Neumann, Mark Strembeck, A Scenario-driven Role Engineering Process for Functional RBAC Roles, SACMAT'02, June 34, 2002, Monterey, California, USA.

[4] Coyne, E. J. 1996. Role engineering. First ACM Workshop on Role-Based Access Control, Gaithersburg, Maryland, United States.

[5] N. Damianou, N. Dulay, E. Lupu, M. Sloman , The Ponder Policy Specification Language Workshop on Policies for Distributed Systems and Networks (Policy2001), HP Labs Bristol, 29-31. Springer-Verlag.

[6] Bertino, E., Mileo, A., and Provetti, A. 2005. PDL with Preferences. IEEE international Workshop on Policies For Distributed Systems and Networks, Policy 2005 – Vol. 00, IEEE Computer Society, Washington, DC, 213-222.

[7] Basile, C.; Lioy, A.; Perez, G. Martinez; C., F. J. Garcia; Skarmeta, A. F. Gomez, POSITIF: A Policy-Based Security Management SystemPolicies for Distributed Systems and Networks, 2007. POLICY'07, pp. 280 – 280.

[8] Lalana Kagal, Rei : A Policy Language for the Me-Centric Project, TechReport, HP Labs, September 2002.

[9] Colin Camerer, Redirecting Research in Business Policy and Strategy, Strategic Management Journal, Vol.6, No. 1. (Jan. – Mar., 1985), pp. 1-15.

[10] René Wies, Using a Classification of Management Policies for Policy Specification and Policy Transformation. In Proc. ISINM '95, Santa Barbara, California, May 1995.

[11] André Rifaut, Christophe Feltus, Improving Operational Risk Management Systems by Formalizing the Basel II Regulation with Goal Models and the ISO/IEC 15504 Approach, REMO2V'2006, Luxembourg.

[12] Davrondhon Gafurov, Kirsi Helkala, Nils Kalstad Svendsen, Security models for electronic medical record, Telektronikk 1.2005.

[13] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn and Ramaswamy Chandramouli, Proposed NIST Standard for Role-Based Access Control, ACM Transactions on Information and System Security, Vol. 4, No. 3, August 2001, Pages 224-274.

[14] C. Bettini, S. Jajodia, X. S. Wang, and D. Wijesekera, Provisions and Obligations in Policy Management and Security Applications, 28th VLDB conference, China, 2002.

[15] Robert Crook, Darrel Ince, Bashar Nuseibeh, Modelling access policies using roles in requirements engineering, Information and Software Technology 45 (2003) 979-991.

[16] Charles B. Haley, Robin C. Laney, Jonathan D. Moffett, and Bashar Nuseibeh, Using Trust Assumptions with Security Requirements, Requirements Engineering Journal, vol. 11 no. 2 (April 2006) pp. 138-15.

[17] Robert Crook, Darrel Ince, Bashar Nuseibeh, On Modelling access policies: Relating Roles to their Organisational Context, RE 2005, Paris.

[18] Pete A. Epstein, Engineering of Role/Permission Assignement, PhD thesis.

[19] Crook, R., Ince, D., and Nuseibeh, B., "Using i* to Model Access Policies: Relating Roles to their Organisational Context", Social Modelling for Requirements Engineering, Giorgini, P., Maiden, N., Mylopoulos, J., and Yu, E., eds., MIT Press, 2006.

[20] P.J. Fontaine, Goal-Oriented Elaboration of Security Requirements. M.S. Thesis, Dept. Computing Science, University of Louvain, June 2001.

[21] Yu, E. S. and Liu, L. 2001. Modelling Trust for System Design Using the i* Strategic Actors Framework. Workshop on Deception, Fraud, and Trust in Agent Societies Held During the Autonomous, Eds. Lecture 35 194.

[22] L. Liu, E. Yu, J. Mylopoulos, Analyzing Security Requirements as Relationships Among Strategic Actors, SREIS'02, Raleigh, North Carolina, 2002.

[23] A. Antón, Goal-Based Requirements Analysi,. Second ICRE'96, Colorado Springs, USA, 1996.

[24] Robert Crook, Darrel Ince, Bashar Nuseibeh, Towards an Analytical Role Modelling Framework for Security Requirements, Security Requirements Group, Departement of Computing, The Open University, Walton Hall, Milton Keynes, MK7 6AA, UK.

[25] Henry Mintzberg, Structure in Fives: Designing Effective Organisations, Englewood Cliffs, NJ: Prentice-Hall, 1983. pp. 312

[26] Qingfeng He, Annies I. Antón, "A Framework for Privacy-Enhanced Access Control Analysis in Requirements Engineering", Proc. of the 9th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'03), pp. 137-146, Klagenfurt/Velden, Austria, June 16-17, 2003.

[27] E. B. Fernandez and J. C. Hawkins, "Determining Role Rights from Use Cases", Proc. of the ACM Workshop on Role-Based Access Control, 1997.

[28] Roeckle, H., Schimpf, G., and Weidinger, R. 2000. Process-oriented approach for role-finding to implement role-based security administration in a large industrial organization. In Proceedings of the Fifth ACM Workshop on Role-Based Access Control (Berlin, Germany, July 26 - 28, 2000). Role-Based Access Control '00.

[29] Chandramouli, R. 2001. A Framework for Multiple Authorization Types in a Healthcare Application System. 17th Annual Computer Security Applications Conference, 2001. ACSAC. IEEE Computer Society, Washington, DC, 137.

[30] D. J. Thomsen, Richard C. O'Brien and C. Payne, Napoleon: Network Application Policy Environment, ACM Workshop on Role-Based Access Control, 1999, pp. 145-152.

[31] N. Dulay, E. Lupu, M. Solman, N. Damianou, A Policy Deployment Model for the Ponder Language , An extended version of paper in Proc. IEEE/IFIP International Symposium on Integrated Network Management, (IM'2001), Seattle, May 2001, IEEE Press.

[32] OASIS, "eXtensible Access Control Markup Language (XACM) Version 2.0" February 2005. www.oasis-open.org/committees/xacml/